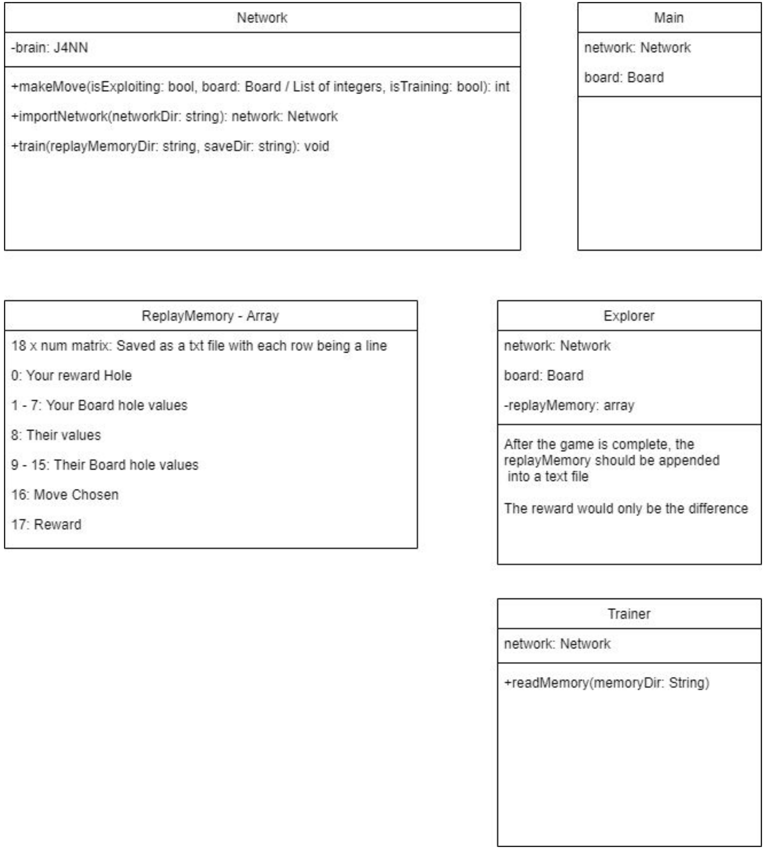
Kalah Bot

Team 16

Method we used

We used Deep Learning

We split it into 4 parts:



Network

-brain: J4NN

+makeMove(isExploiting: bool, board: Board / List of integers, isTraining: bool): int

+importNetwork(networkDir: string): network: Network

+train(replayMemoryDir: string, saveDir: string): void

Main

network: Network

board: Board

ReplayMemory - Array

18 x num matrix: Saved as a txt file with each row being a line

0: Your reward Hole

1 - 7: Your Board hole values

8: Their values

9 - 15: Their Board hole values

16: Move Chosen

17: Reward

Explorer

network: Network

board: Board

-replayMemory: array

After the game is complete, the replayMemory should be appended into a text file

The reward would only be the difference

Trainer

network: Network

+readMemory(memoryDir: String)

Network Creation and Testing

- The network has has the input and output size fixed
- The hidden networks node size and number were tweaked
- Testing stages and updating of both training and target networks

Explorer Architecture

❏ The Explorer class is used for training by playing the game

❏ Using the Q learning algorithm

❏ The gaming history is stored in a matrix called ‘replayMemory’, which will be passed on to the Brain through the Network class

❏ Updating the Q value for all the states and their actions at the end of the game

Network Architecture

- Main functionality:
  - Connector for the Brain and the player (Main/Explorer)
  - Create a new Brain object
  - Return a move for Main and Explorer.
  - Using Epsilon Greedy Algorithm to decide move for the Explorer
  - Translation from board to string for the Brain to predict the best move

Pie Rule (Plan A)

Play agent against itself 100 times for each opening move

Find out which side wins most often

Set up swap to play the side with the most wins

Idea being it is the side our agent prefers to play for each opening

Pie Rule (Plan B)

Use heuristics and a board evaluation function to assess opening positions and response moves (Difference of pieces in pits, difference of pieces between the sides, momentum move opportunity, empty pits with large amount on other side)

For start move 1, north is better so don’t swap (momentum move)

For start moves 2,3 and 4, board position equal so swap so you have next move

For start moves 5,6 and 7, north slightly better so don't swap (more pieces on his side)

University of San Francisco course: https://usfca.instructure.com/courses/1553625/assignments/6549876

Jeffrey Popyack Case Study: https://www.cs.drexel.edu/~jpopyack/Courses/AI/Sp15/notes/CaseStudy_Kalah.pdf

What we have completed

Completed the architecture of the bot

Running in a highly specified environment

We have trained the network for about a day

The current file size is quite large (1.1GB)

What we expect to accomplish by submission

Reduce the file size (Unneded data and file paths)

Try to compile for a less specific set-up

Train for a longer period